,Міністерство освіти і науки України Вінницький національний технічний університет

Факультет інформаційних технологій та комп‘ютерної інженерії Кафедра обчислювальної техніки

КУРСОВА РОБОТА

з дисципліни «КОМП‘ЮТЕРНА ЛОГІКА»

на тему: «СИНТЕЗ ЦИФРОВОГО АВТОМАТА З ЖОРСТКОЮ ЛОГІКОЮ»

Студента 2 курсу групи 2КІ-21Б спеціальності «123 Комп‘ютерна інженерія»

Самуся Олександра Віталійовича \_\_\_\_\_\_

Керівник : Богомолов С.В.

Кількість балів: Оцінка: ECTS

Члени комісії \_І.С. Колесник

(підпис) (прізвище та ініціали)

(підпис) (прізвище та ініціали)

м. Вінниця — 2022

Міністерство освіти і науки України Вінницький національний технічний університет

2

Факультет інформаційних технологій та комп‘ютерної інженерії

ЗАТВЕРДЖУЮ

Зав. кафедри ОТ, д.т.н., проф.

О. Д. Азаров (підпис)

«19» вересня 2022 р.

ЗАВДАННЯ

на курсову роботу з дисципліни «Комп‘ютерна логіка» студенту Самусю Олександру Віталійовичу групи 2КІ—21б варіант 16 Здійснити синтез цифрового автомата з жорсткою логікою. Вихідні дані:

1. Таблиця переходів та виходів:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a1 | a2 | a3 | a4 |  |  | a1 | a2 | a3 | a4 |
| z1 | a1 | a4 | a3 | — | z1 | w2 | w2 | w3 | — |
| z2 | a2 | a3 | a2 | a1 | z2 | w1 | w4 | w3 | w2 |

1. Представити автомат суміщеною таблицею та графом.
2. Тип керуючого автомата: Довільно.
3. Тип елементів пам‘яті: D -тригер.
4. Виконати перехід до аналітичного представлення функцій.
5. Виконати мінімізацію функцій або їх заперечення методом: Квайна Мак Класкі
6. Представити ці функції в усіх восьми нормальних формах (базисах).
7. Обрати операторні форми, що забезпечують одержання комбінаційних схем з мінімальною кількістю використаних мікросхем та максимальною швидкодією. Побудувати зазначені схеми на елементах: 3ТА-НІ, 2АБО (кількість у корпусі — 2, 4 відповідно; час затримки — 24 нс, 22 нс відповідно).
8. Здійснити моделювання автомата.

Графічна частина: схеми структурна, функціональна, електрична принципова на елементах вибраної серії мікросхем та графіки перехідних процесів.

Дата видачі завдання «\_19\_» вересня 2022 р.

Завдання до виконання прийняв

«19»\_вересня 2022 р.

Керівник: Богомолов С.В.

# АНОТАЦІЯ

3

У курсовій роботі представлено синтез цифрового автомата з жорсткою логікою. Розглянуто теоретичні основи синтезу цифрових автоматів та здійснено синтез цифрового автомата відповідно до завдання, а саме побудова автомата Мілі на базі D-тригера, мінімізацію здійснено методом Квайна Мак Класкі схему виконано в базисі ТА-НІ/АБО. Проведено моделювання в додатку MicroCap 12 та створено структурну, функціональну та електричну принципові схеми.

# ЗМІСТ

[ВСТУП 4](#_TOC_250012)

* 1. [СИНТЕЗ ЦИФРОВОГО АВТОМАТА З ЖОРСТКОЮ ЛОГІКОЮ 5](#_TOC_250005)
     1. [Абстрактний синтез 5](#_TOC_250004)
     2. [Структурний синтез 6](#_TOC_250003)
        1. Аналітичне представлення функцій 6
        2. Мінімізація функцій 8
        3. Представлення функцій в різних елементних базисах 18
        4. Вибір елементної бази 20
        5. Аналіз та розрахунок параметрів 21
     3. [Моделювання 23](#_TOC_250002)

[ВИСНОВКИ 25](#_TOC_250001)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 26](#_TOC_250000)

ДОДАТОК А — Схема структурна 27

ДОДАТОК Б — Схема функціональна 28

ДОДАТОК В — Схема електрична принципова 29

# ВСТУП

Комп‘ютерна логіка — це навчальна дисципліна, що охоплює логічні, математичні та технічні основи, базові принципи, поняття та моделі обчислювальних та управляючих цифрових систем. Назва «комп‘ютерна логіка» може вживатися у широкому та вузькому розумінні. У широкому розумінні комп‘ютерна логіка охоплює собою всю множину логік, закономірностей, моделей та методів функціонування комп‘ютерних систем та їх компонент. За такого трактування комп‘ютерної логіки слушно пов‘язати всі ці логіки із різними рівнями ієрархічної організації комп‘ютерної системи.

Комп‘ютерна логіка досліджує ієрархію моделей, що описують фундаментальні закономірності функціонування комп‘ютерних систем як програмно-апаратних засобів різної складності та призначення. Комп‘ютерна логіка вивчає моделі, методи аналізу та синтезу логічних комбінаційних схем, цифрових автоматів, що є базовими логічними моделями сучасних комп‘ютерних систем та систем контролю і управління. Основним її завданням є роз‘яснення того, як розуміє та обробляє інформацію . комп‘ютер.

У вузькому, більш прикладному, розумінні поняття «комп‘ютерна логіка» охоплює собою множину питань, що стосуються низькорівневих аспектів, логік, моделей функціонування обчислювальних систем та засобів автоматизації і управління.

Фундаментальною логічною моделлю комп‘ютерної системи є поняття автомату. По-перше, автомат — це пристрій (система), який без участі людини здійснює відбір, опрацювання, передавання та зберігання інформації (даних, повідомлень, сигналів) відповідно до закладеного у нього алгоритму. З іншого боку, автомат — це математична модель реальної технічної системи (наприклад, комп‘ютера чи певного його вузла, елемента). У цьому випадку автомат розглядається як деяка «чорна скринька», що має скінченну кількість входів та виходів, а також деяку множину внутрішніх станів.

# СИНТЕЗ ЦИФРОВОГО АВТОМАТА З ЖОРСТКОЮ ЛОГІКОЮ

## Абстрактний синтез

Таблиці переходів та виходів цифрового автомату (табл. 2.1 та 2.2):

|  |  |
| --- | --- |
| Таблиця 1.1 — Таблиці переходів | Таблиця 1.2 — Таблиці виходів |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a1 | a2 | a3 | a4 |  |  | a1 | a2 | a3 | a4 |
| z1 | a1 | a4 | а3 | — | z1 | w2 | w2 | w3 | — |
| z2 | a2 | a3 | a2 | а1 | z2 | w1 | w4 | w3 | w2 |

Функція переходів та функція виходів автомата Мілі можуть бути задані суміщеною таблицею табл. 2.3.

Таблиця 2.3 — Суміщена таблиця автомата Мілі

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a1 | a2 | a3 | a4 |
| z1 | z1 / w2 | z1 / w2 | z1 / w3 | — |
| z2 | z2 / w1 | z2 / w4 | z2 / w3 | z2 / w2 |

Автомат Мілі може бути заданий графом. При графічному способі завдання автомата здійснюється за допомогою графу.

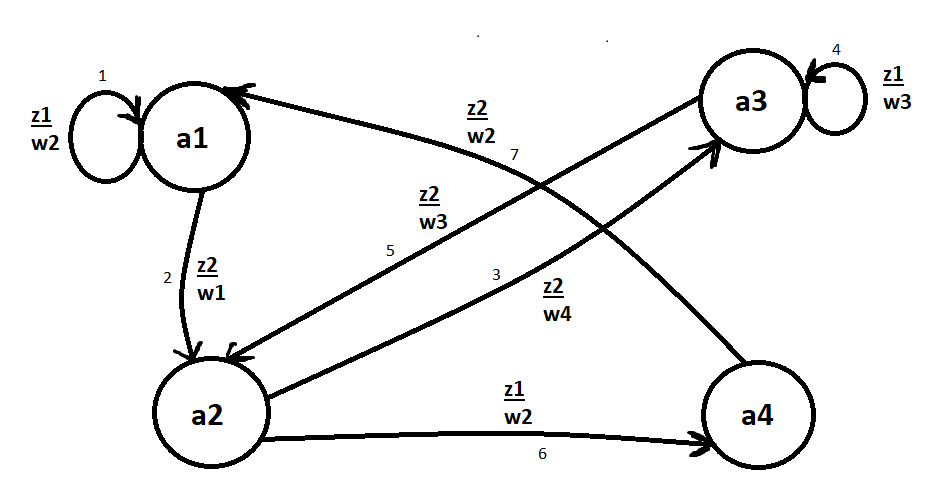


Рисунок 1.1 — Граф автомата Мілі

## Структурний синтез

* + 1. Аналітичне представлення функцій

Оберемо в якості елементів пам’яті *D-тригер*.

Таблиця 1.4 — Кодування вхідних станів автомата

Таблиця 1.5 — Кодування вхідних станів автомата

Таблиця 1.6 — Кодування переходів автомата

|  |  |  |
| --- | --- | --- |
|  | x1 | x2 |
| z1 | 0 | 0 |
| z2 | 0 | 1 |
| z3 | 1 | 1 |
| z4 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
|  | y1 | y2 |
| w1 | 0 | 0 |
| w2 | 0 | 1 |
| w3 | 1 | 1 |
| w4 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
|  | Q1 | Q2 |
| a1 | 0 | 0 |
| a2 | 0 | 1 |
| a3 | 1 | 1 |
| a4 | 1 | 0 |

На основі таблиць закодуємо таблиці виходів та переходів автомата, заданого відповідно до завдання.

Таблиця 1.7 — Кодована таблиця переходів(D1, D2)

Таблиця 1.8 — Кодована таблиця виходів(y1, y2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Q1Q2 | a1 | a2 | a3 | a4 |
|  | x1 x2 | **00** | **01** | **11** | **10** |
| z1 | **00** | 00 | 10 | 11 | — |
| z2 | **01** | 01 | 11 | 01 | 00 |
| z3 | **11** | — | — | — | — |
| z4 | **10** | — | — | — | — |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | y1 y2 | a1 | a2 | a3 | a4 |
|  | x1 x2 | **00** | **01** | **11** | **10** |
| z1 | **00** | 01 | 01 | 11 | — |
| z2 | **01** | 00 | 10 | 11 | 01 |
| z3 | **11** | — | — | — | — |
| z4 | **10** | — | — | — | — |

При канонічному методі синтез зводиться до одержання системи функцій:

Та наступної побудови комбінаційних схем, що реалізують дану систему булевих функції.

Отримаємо фунції y1 та y2 відповідно до завдання.

Функції y1 та y2 отримані з таблиці виходів у вигляді ДДНФ:

Таблиця 1.9 — Кодована таблиця Таблиця 1.10 — Кодована таблиця

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q1Q2  x1x2 | **00** | **01** | **11** | **10** |
| **00** | 0 | 1 | 1 | — |
| **01** | 0 | 1 | 0 | 0 |
| **11** | — | — | — | — |
| **10** | — | — | — | — |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q1Q2  x1x2 | **00** | **01** | **11** | **10** |
| **00** | 0 | 0 | 1 | — |
| **01** | 1 | 1 | 1 | 0 |
| **11** | — | — | — | — |
| **10** | — | — | — | — |

На основі даних таблиць 1.9-1.10 сформуємо функції D1 та D2:

* + 1. Мінімізація функцій

Мінімальною формою представлення логічної функції називають таку форму, яка не допускає більше ніяких спрощень. При мінімізації виходять з вимоги мінімальної кількості логічних елементів, реалізованих у вигляді мікросхем.

Для мінімізації логічних функцій використовуються різні методи: послідовного виключення змінних з допомогою законів та тотожностей алгебри логіки.

Таблиця 1.11 — Кодована таблиця D1, D2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  набору | X1 | X2 | Q1 | Q2 | D1 | D2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | — | — |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 0 | 1 |
| 8 | 1 | 0 | 0 | 0 | — | — |
| 9 | 1 | 0 | 0 | 1 | — | — |
| 10 | 1 | 0 | 1 | 0 | — | — |
| 11 | 1 | 0 | 1 | 1 | — | — |
| 12 | 1 | 1 | 0 | 0 | — | — |
| 13 | 1 | 1 | 0 | 1 | — | — |
| 14 | 1 | 1 | 1 | 0 | — | — |
| 15 | 1 | 1 | 1 | 1 | — | — |

Відповідно до завдання оберемо метод мінімізації за Квайном Мак Класкі.

При мінімізації функцій вважаємо що невизначені стани кодуються 1.

Мінімізація D1:

Для функції D1 проведемо операцію склеювання, представимо її у вигляді таблиці 1.12:

Таблиця 1.12 — Склейки функцій

|  |  |  |
| --- | --- | --- |
| 0001 | x - - -  -001  -011  -101 | -0-1\*  --01\* |
| 0011  0101  1001 | - x - -  0-01  1-01 | --01 |
| 1011  1101 | - - x -  00-1  10-1 | -0-1 |

Утворимо таблицю покриття для функції D1(табл. 1.13): Таблиця 1.13 — Таблиця покриття для функції D1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0001 | 0011 | 0101 | 1001 | 1011 | 1101 |
| -0-1 | V | V | — | V | V | — |
| --01 | V | — | V | V | — | V |

В результаті мінімізації для функції D1 маємо таку форму:

Проведемо аналогічну операцію для мінімізації зворотної вітки функції, але під час склеювання використаємо конституенти нуля. Записуємо конституенти у групи відповідно до кількості одиниць у табл. 1.13.

Таблиця 1.13 — Склейки функцій

|  |  |  |
| --- | --- | --- |
| 0000 | x - - -  -000  -010  -100  -110  -111 | -0-0  --00 ---0\*  --10  -1-0  -11- |
| 0010  0100  1000 | - x - -  0-00  0-10  1-00  1-10 | 0--0  --00  --10  1--0 |
| 0110  1100  1010 | - - x -  00-0  01-0  10-0  11-0 | 0--0  -0-0  -1-0  1--0 |
| 0111  1110 | - - - x  011-  111- | -11-\* |
| 1111 |  |  |

Утворимо таблицю покриття для функції(табл. 1.14): Таблиця 1.14 — Таблиця покриття

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0000 | 0010 | 0100 | 1000 | 0110 | 1100 | 1010 | 1110 | 0111 | 1111 |
| -11- | — | — | — | — | V | — | — | V | V | V |
| ---0 | V | V | V | V | V | V | V | V | — | — |

В результаті мінімізації для функції маємо таку форму:

Мінімізація D2:

Для функції D2 проведемо операцію склеювання, представимо її у вигляді таблиці 1.15:

Таблиця 1.15 — Склейки функцій

|  |  |  |
| --- | --- | --- |
| 0100 | x - - -  -100  -011  -101  -111 | -10-  --11 |
| 0011  0101  1100 | - x - -  0-11  1-11 | --11\* |
| 0111  1011  1101 | - - x -  01-1  11-1 | -1-1 |
| 1111 | - - - x  010-  110- | -10-\* |

Утворимо таблицю покриття для функції D2 (табл. 1.16): Таблиця 1.16 — Таблиця покриття для функції D2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0000 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 |
| -10- | — | V | V | — | — | V | V | — |
| --11 | V | — | — | V | V | — | — | V |

У результаті мінімізації для функції D2 маємо таку форму:

Проведемо аналогічну операцію для мінімізаціцї зворотної вітки

функції, але під час склеювання використаємо конституєнти нуля. Записуємо конституенти у групи відповідно до кількості у табл. 1.17

Таблиця 1.17 — Склейки функцій

|  |  |  |
| --- | --- | --- |
| 0000 | x - - -  -000  -001  -010  -110 | -00-  -0-0  --10 |
| 0001  0010  1000 | - x - -  0-10  1-10 | --10\* |
| 0110  1001  1010 | - - x -  00-0  10-0 | -0-0\* |
| 1110 | - - - x  000-  100- | -00-\* |

Утворимо таблицю покриття для функції(табл. 1.18):

Таблиця 1.18 — Таблиця покриття для функції

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0000 | 0001 | 0010 | 1000 | 0110 | 1001 | 1010 | 1110 |
| -00- | V | V | — | V | — | V | — | — |
| -0-0 | V | — | V | V | — | — | V | — |
| --10 | — | — | V | — | V | — | V | V |

В результаті мінімізації для функції маємо таку форму:

Утворимо кодовану таблицю (табл 1.19) і на основі кодованих таблиць виходів і застосуємо мінімізацію за Квайном Мак Класкі, відповідно до завдання.

Кодована таблиця

Таблиця 1.19

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  набору | X1 | X2 | Q1 | Q2 | у1 | у2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 1 | 0 |
| 14 | 1 | 1 | 1 | 0 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 1 | 1 |

Замінюємо всі конституенти одиниці їх двійковим представленням, далі записуємо конституенти одиниці у групи відповідно до кількості.

Мінімізація y1:

Для функції у1 проведемо операцію склеювання, представимо її у вигляді таблиці 1.20

Таблиця 1.20 — Склейки функцій y1

|  |  |  |
| --- | --- | --- |
| 0011  0101 | x - - -  -011  -101  -111 | --11  -1-1 |
| 0111  1011  1101 | - x - -  0-11  1-11 | --11\* |
| 1111 | - - x -  01-1  11-1 | -1-1\* |

Утворимо таблицю покриття для функції у1 (табл. 1.21): Таблиця 1.21 — Таблиця покриття для функції у1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0011 | 0101 | 0111 | 1011 | 1101 | 1111 |
| --11 | V | — | V | V | — | V |
| -1-1 | — | V | V | — | V | V |

В результаті мінімізації для функції маємо таку форму:

Проведемо аналогічну операцію для мінімізаціцї зворотної вітки функції, але під час склеювання використаємо конституєнти нуля. Записуємо конституенти у групи відповідно до кількості у табл. 1.22

Таблиця 1.22 — Склейки функцій

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 | x - - -  -000  -001  -010  -100  -110 | -00-\*  -0-0  --00  --10  -1-0 | ---0\* |
| 0001  0010  0100  1000 | - x - -  0-00  0-10  1-00  1-10 | 0--0  --00  --10  1--0 | ---0 |
| 0110  1001  1010  1100 | - - x -  00-0  01-0  10-1  11-0 | 0--0  -1-0 |  |
| 1110 | - - - x  000-  100- | -00- |  |

Утворимо таблицю покриття для функції(табл. 1.23):

Таблиця 1.23 — Таблиця покриття для функції

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0000 | 0001 | 0010 | 0100 | 1000 | 0110 | 1001 | 1010 | 1100 | 1110 |
| ---0 | V | — | V | V | V | V | — | V | V | V |
| -00- | V | V | — | — | V | — | V | — | — | — |

В результаті мінімізації для функції маємо таку форму:

Мінімізація y2

Для функції у2 проведемо операцію склеювання, представимо її у вигляді таблиці 1.24:

Таблиця 1.24 — Склейки функцій y2

|  |  |  |
| --- | --- | --- |
| 0000 | x - - -  -000  -001  -010  -011  -110  -111 | -00-  -01-  -11- |
| 0001  0010  1000 | - x - -  0-10  0-11  1-10  1-11 | 0-1-  1-1- --1-\* |
| 0011  0110  1001  1010 | - - x –  00-0  00-1  10-0  10-1 | 00--  10-- -0--\* |
| 0111  1011  1110 | - - - x  000-  001-  100-  011-  101-  111- | 00--  1-1- |
| 1111 |  |  |

Утворимо таблицю покриття для функції у2 (табл. 1.24): Таблиця 1.24 — Таблиця покриття для функції у2

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0000 | 0001 | 0010 | 0011 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1110 | 1111 |
| --1- | — | — | V | V | V | V | — | — | V | V | V | V |
| -0-- | V | V | V | V | — | — | V | V | V | V | — | — |

В результаті мінімізації для функції маємо таку форму:

Проведемо аналогічну операцію для мінімізації зворотної вітки функції, але під час склеювання використаємо конституенти нуля. Записуємо конституенти у групи відповідно до кількості у табл. 1.25

Таблиця 1.25 — Склейки функцій

|  |  |  |
| --- | --- | --- |
| 0100 | x - - -  -100  -101 | -10-\* |
| 0101  1100 | - x- - |  |
| 1101 | - - x - |  |
|  | - - - x  010-  110- | -10- |

Утворимо таблицю покриття для функції у2 (табл. 1.25): Таблиця 1.25 — Таблиця покриття для функції

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0100 | 0101 | 1100 | 1101 |
| -10- | V | V | V | V |

В результаті мінімізації для функції маємо таку форму:

В результаті всіх мінімізації маємо такі функції для представлення прямих віток автомата:

Для зворотних віток функцій автомата маємо такі представлення:

Для моделювання автомата використаємо мінімізовані функції, представивши їх у 8 елементних базисах.

* + 1. Представлення функцій в різних елементних базисах

Пряма вітка для D1:

Пряма вітка для D2:

Пряма вітка для y1:

Пряма вітка для y2:

Інверсна вітка для :

Інверсна вітка для :

Інверсна вітка для :

Інверсна вітка для :

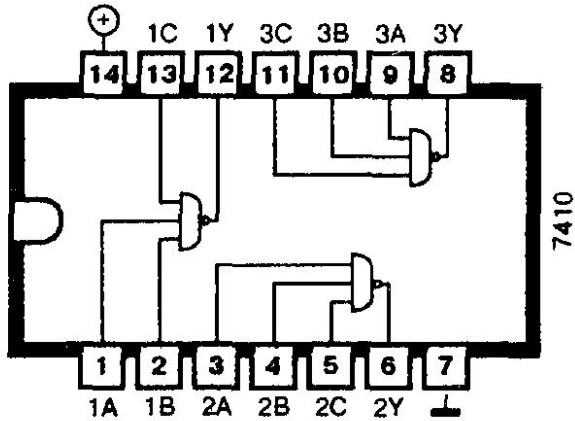
* + 1. Вибір елементної бази

В якості елементів пам‘яті оберемо D-тригер. Оскільки схему потрібно побудувати на елементах 3ТА-НІ та 2АБО, тоді оберемо базис ТА-НІ/АБО.

Для реалізації відповідного автомата використаємо такі мікросхеми:

—7410 (3ТА-НІ);

—7432 (2АБО).

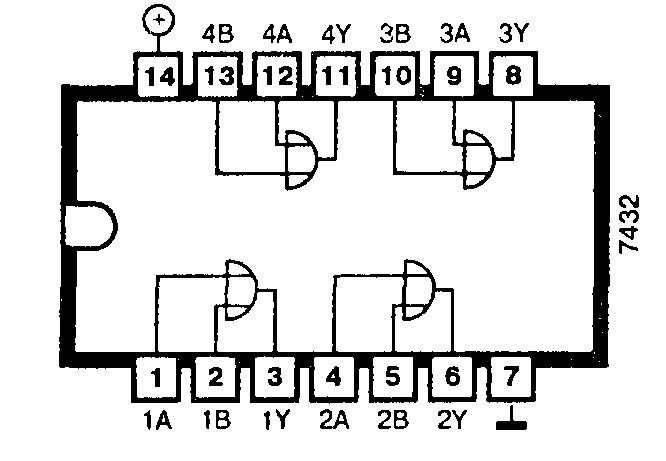


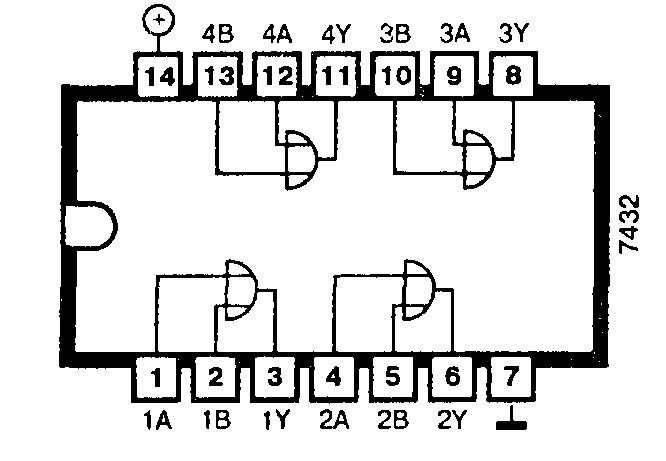
Малюнок 1.2 — Мікросхема 7410

Мікросхема 7410 містить три окремі логічні елементи ТА-НІ, у кожного з яких по три входи. Застосовується для реалізації логічних функцій Та, ТА-НІ, інвертування сигналів.

Мікросхема 7432 містить чотири логічні елементи АБО, з двома виходами на кожному. Всі чотири логічні елементи можна використовувати незалежно один від одного.

Використовується для реалізації логічної функції АБО.





Малюнок 1.3 — Мікросхема 7432

* + 1. Аналіз та розрахунок параметрів

1) Складність за Квайном-Мак-Класкі

D1 = 4 + 2 \* 2 = 8

D2 = 3 + 3 \* 2 = 9

y1 = 2 + 4 \* 2 = 10

y2 = 3 + 3 \* 2 = 9

2) Кількість логічних елементів

M = 12

3) Усереднене значення часу затримки

4) Середній час затримки сигналів

T1 = 18.5 + 18.5 = 37 (нс)

T2 = 18.5 (нс)

y1 = 18.5 (нс)

y2 = 18.5 + 18.5 = 37 (нс)

T1\* = 15 + 22 = 37 (нс)

T2\* = 22 (нс)

y1\* = 22 (нс)

y2\* = 15 + 22 = 37 (нс)

5) Споживана потужність мікросхем

6) Загальна активна потужність схем

T1 = 3 \* 2 + 5 = 11 (Вт)

T2 = 3 (Вт)

y1 = 3 (Вт)

y2 = 3 (Вт)

1.3 Моделювання

На основі графа(рисунок 1.1) запрограмуємо генератор сигналів, описуючи кожну комбінацію.

Виберемо початкову вершину a1, з якої і будемо починати

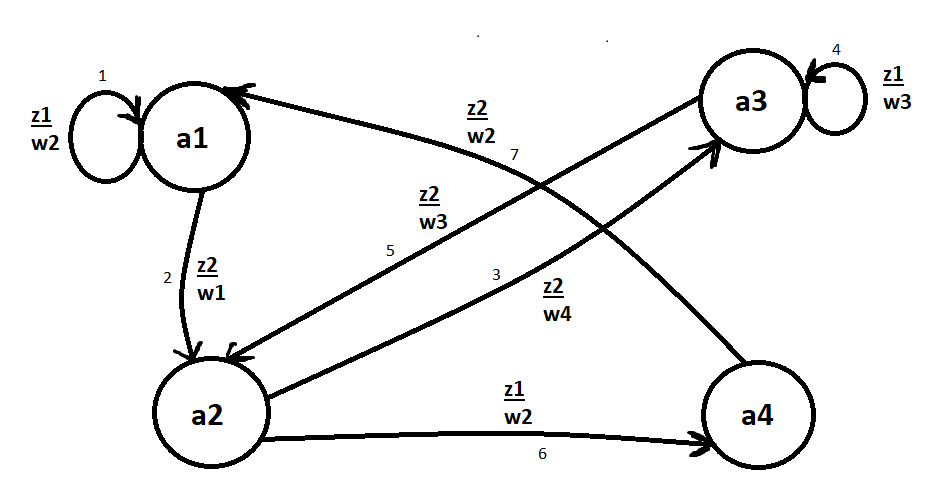


Рисунок 1.1 – Граф автомата Мілі

В результаті отримаємо наступний код генератора:

.define Count2

+label=start

+0n 00

+100n 01

+200n 01

+300n 00

+400n 01

+500n 00

+600n 01

+700n goto start -1 times

На основі цих даних побудуємо функціональну схему в середовищі MicroCap12(рисунок 1.2)

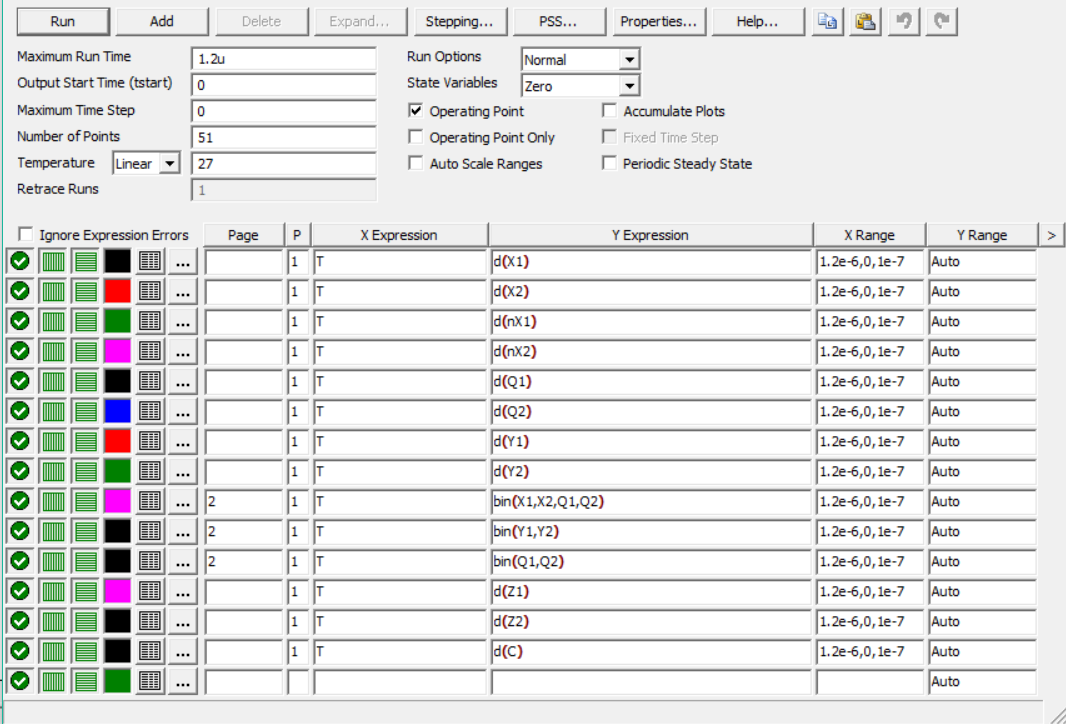
**

Рисунок 1.2 – Налаштування аналізу

Результати моделювання(рисунок 1.3)

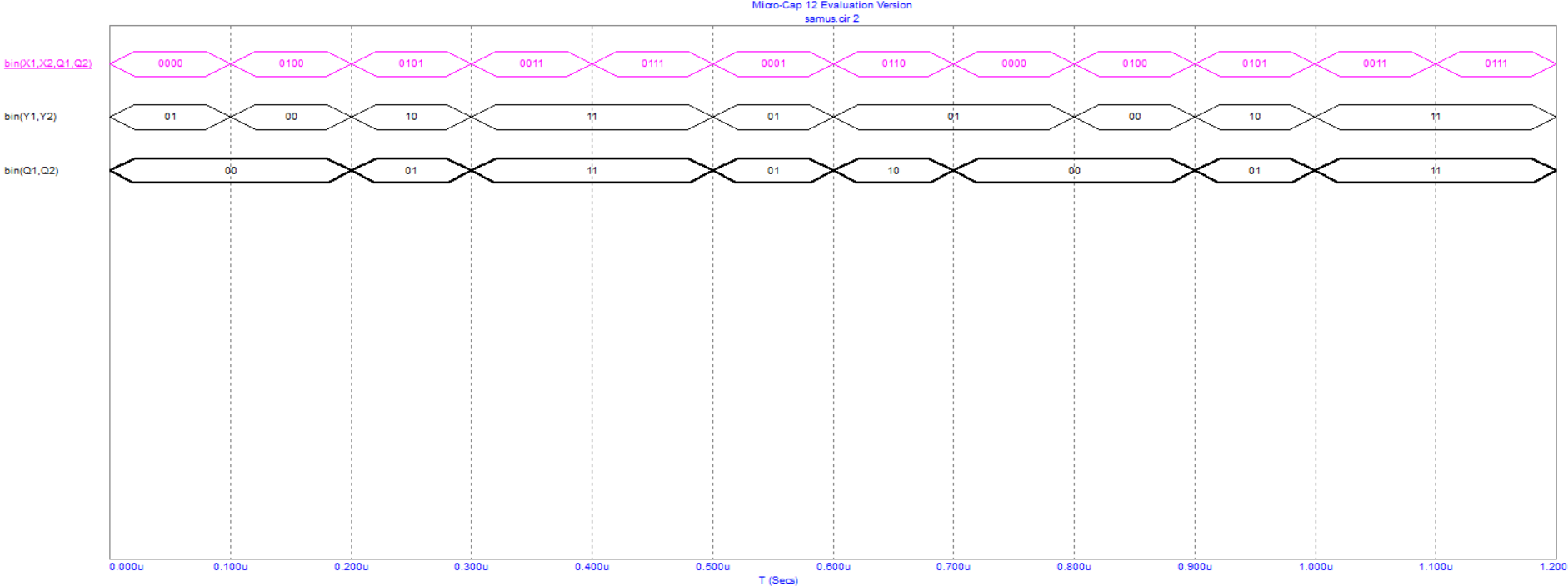


Рисунок 1.3

**Висновки**

Під час виконання курсового проекту було розглянуто принцип роботи цифрового автомата з жорсткою логікою. Були розглянуті теоретичні основи і поняття теорії цифрового автомату Мілі, його особливості і використання.

Було досліджено принципи роботи автомату Мілі і D-тригеру, який дозволив виконати моделювання і розрахунки в середовищі MicroCap12.

Також було проведено абстрактний синтез автомата, виконане графічне представлення автомата, і його представлення у вигляді суміщеної таблиці і у вигляді функцій.

Під час проведення структурного синтезу автомата функції, яка описує його роботу, було мінімізовано за допомогою метода Квайна-Мак-Класкі. Для того, щоб змоделювати автомат, потрібно було виконати моделювання у 8 нормальних формах(базисах), також було проведено дослідження і аналіз елементної бази.

Результат моделювання повністю збігається з розрахунками, які були проведені під час роботи.

**Джерела**

1) Мельник А.О. "Архітектура комп‘ютера". Наукове видання. Луцьк. Волинська обласна друкарня, 2008. 470 с

2) Жабін В. І., Жуков І. А. Прикладна теорія цифрових автоматів. Київ: НАУ-Друк, 2009. 360 с

3) Електронне мережне навчальне видання Дичка І. А., Легеза В.П., Онай М. В. Комп’ютерна логіка. Прикладна теорія цифрових автоматів. КПІ ім. Ігоря Сікорського

Додаток А

Схема комбінаційна

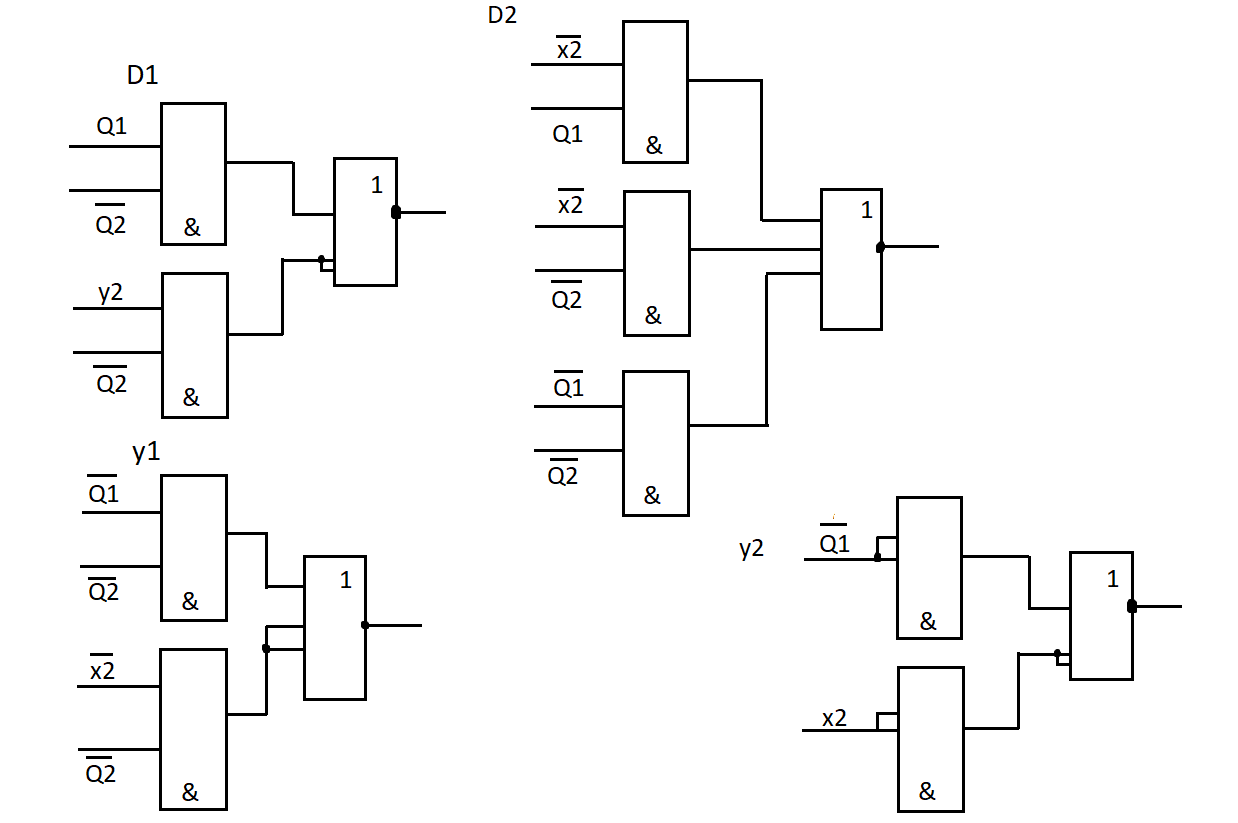


Схема функціональна

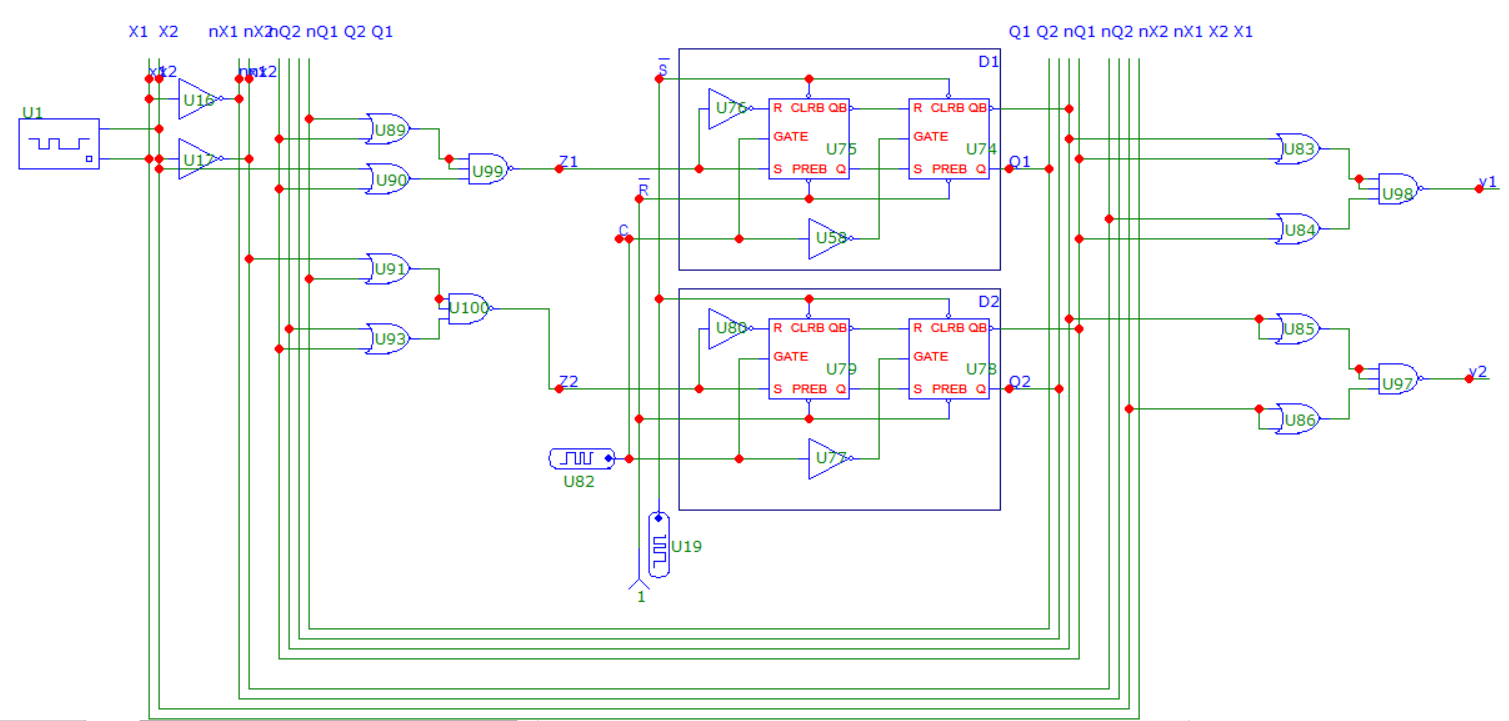
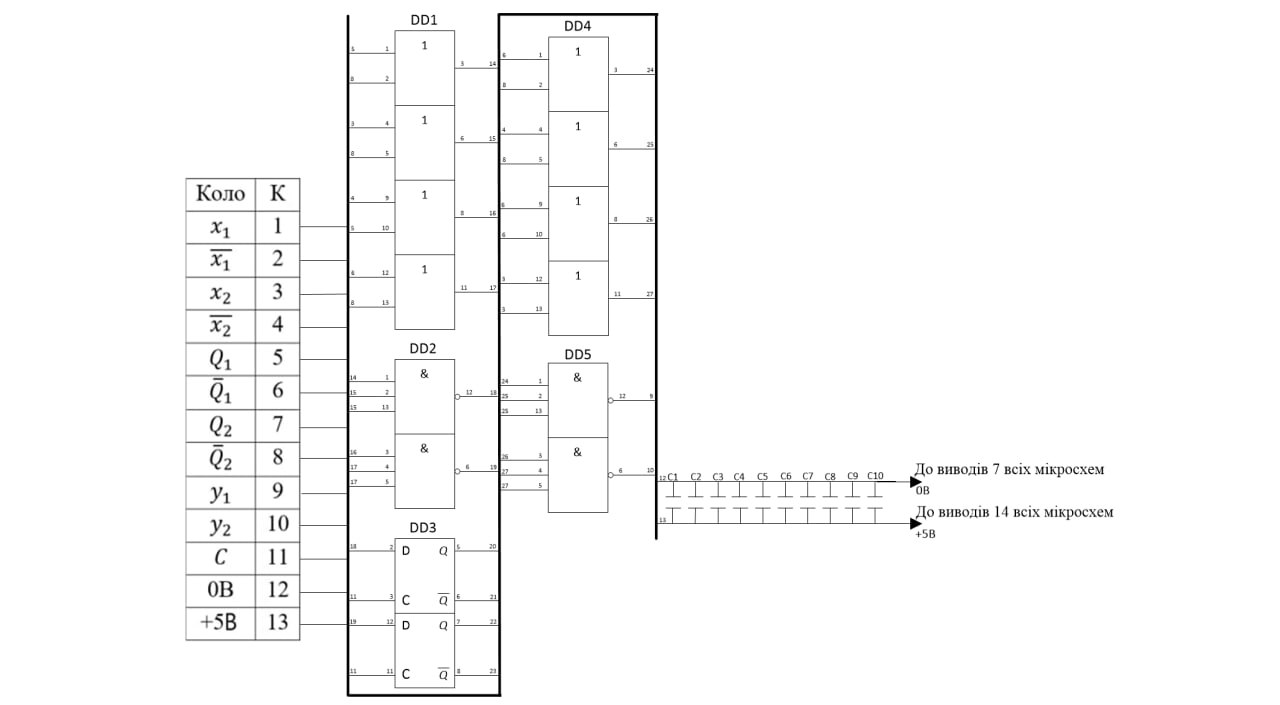


Схема електрична принципова



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Поз.  познач. | | Найменування | | | | Кіл | | Примітки | | |
|  | | Мікросхеми | | | |  | |  | | |
| DD1,DD2,DD3 | | 7432 | | | | 3 | |  | | |
| DD4,DD5 | | 7410 | | | | 2 | |  | | |
|  | |  | | | |  | |  | | |
|  | | Конденсатори | | | |  | |  | | |
| С1…С10 | | 0805-Х7R-0,1 uF+5% - 50V | | | | 10 | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  | |  | | | |  | |  | | |
|  |  |  |  |  | *08-23.КЛ.0016.00.000 ПЕЗ* | | | | | |
|  |  |  |  |  |
| Зм. | Арк | № докум. | Підпис |  |
| Розробив | | Самусь О.В. |  |  | Перетворення булевої функції та синтез комбінаційної схеми  Перелік елементів | | Літ. | | Арк | Аркушів |
| Перевірив | | Богомолов С.В. |  |  |  | | 1 | 1 |
|  | |  |  |  | 2КІ-21б | | | |
| Н. контр. | |  |  |  |